1. Welcome to the world of data science

Throughout the world of data science, there are many languages and tools that can be used to complete a given task. While you are often able to use whichever tool you prefer, it is often important for analysts to work with similar platforms so that they can share their code with one another. Learning what professionals in the data science industry use while at work can help you gain a better understanding of things that you may be asked to do in the future.

In this project, we are going to find out what tools and languages professionals use in their day-to-day work. Our data comes from the Kaggle Data Science Survey (https://www.kaggle.com/kaggle/kaggle-survey-2017?utm_medium=partner&utm_source=datacamp.com&utm_campaign=ml+survey+case+study) which includes responses from over 10,000 people that write code to analyze data in their daily work.

In [22]:
```
library(tidyverse)
responses <- read_csv("datasets/kagglesurvey.csv")

responses%>%
head(n=10)
```

```
Parsed with column specification:
cols(
  Respondent = col_double(),
  WorkToolsSelect = col_character(),
  LanguageRecommendationSelect = col_character(),
  EmployerIndustry = col_character(),
  WorkAlgorithmsSelect = col_character()
)
```

A tibble: 10 x 5

Respondent	WorkToolsSelect	La
<dbl>	<chr>	
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	
2	Amazon Machine Learning,Amazon Web services,Cloudera,Hadoop/Hive/Pig,Impala,Java,Mathematica,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft SQL Server Data Mining,NoSQL,Python,R,SAS Base,SAS JMP,SQL,Tableau	
3	C/C++,Jupyter notebooks,MATLAB/Octave,Python,R,TensorFlow	
4	Jupyter notebooks,Python,SQL,TensorFlow	
5	C/C++,Cloudera,Hadoop/Hive/Pig,Java,NoSQL,R,Unix shell / awk	
6	SQL	
7	Jupyter notebooks,NoSQL,Python,R,SQL,Unix shell / awk	
8	Python,Spark / MLlib,Tableau,TensorFlow,Other	
9	Jupyter notebooks,MATLAB/Octave,Python,SAS Base,SQL	
10	C/C++,IBM Cognos,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft R Server (Formerly Revolution Analytics),Microsoft SQL Server Data Mining,Perl,Python,R,SQL,Unix shell / awk	

2. Using multiple tools

Now that we have loaded in the survey results, we want to focus on the tools and languages that the survey respondents use at work.

To get a better idea of how the data are formatted, we will look at the first respondent's tool-use and see that this survey-taker listed multiple tools that are each separated by a comma. To learn how many people use each tool, we need to separate out all of the tools used by each individual. There are several ways to complete this task, but we will use `str_split()` from `stringr` to separate the tools at each comma. Since that will create a list inside of the data frame, we can use the `tidyr` function `unnest()` to separate each list item into a new row.

In [24]:
```
responses[1,2]

tools <- responses  %>%
    mutate(work_tools = str_split(WorkToolsSelect, ","))  %>%
    unnest(work_tools)


head(tools)
```

A tibble: 1 x 1

WorkToolsSelect <chr>
Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl

A tibble: 6 x 6

Respondent <dbl>	WorkToolsSelect <chr>
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl
2	Amazon Machine Learning,Amazon Web services,Cloudera,Hadoop/Hive/Pig,Impala,Java,Mathematica,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft SQL Server Data Mining,NoSQL,Python,R,SAS Base,SAS JMP,SQL,Tableau
2	Amazon Machine Learning,Amazon Web services,Cloudera,Hadoop/Hive/Pig,Impala,Java,Mathematica,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft SQL Server Data Mining,NoSQL,Python,R,SAS Base,SAS JMP,SQL,Tableau
2	Amazon Machine Learning,Amazon Web services,Cloudera,Hadoop/Hive/Pig,Impala,Java,Mathematica,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft SQL Server Data Mining,NoSQL,Python,R,SAS Base,SAS JMP,SQL,Tableau

3. Counting users of each tool

Now that we've split apart all of the tools used by each respondent, we can figure out which tools are the most popular.

In [26]:
```
tool_count <- tools  %>%
    group_by(work_tools)  %>%
    summarise(count = n()) %>%
    arrange(desc(count))

head(tool_count)
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

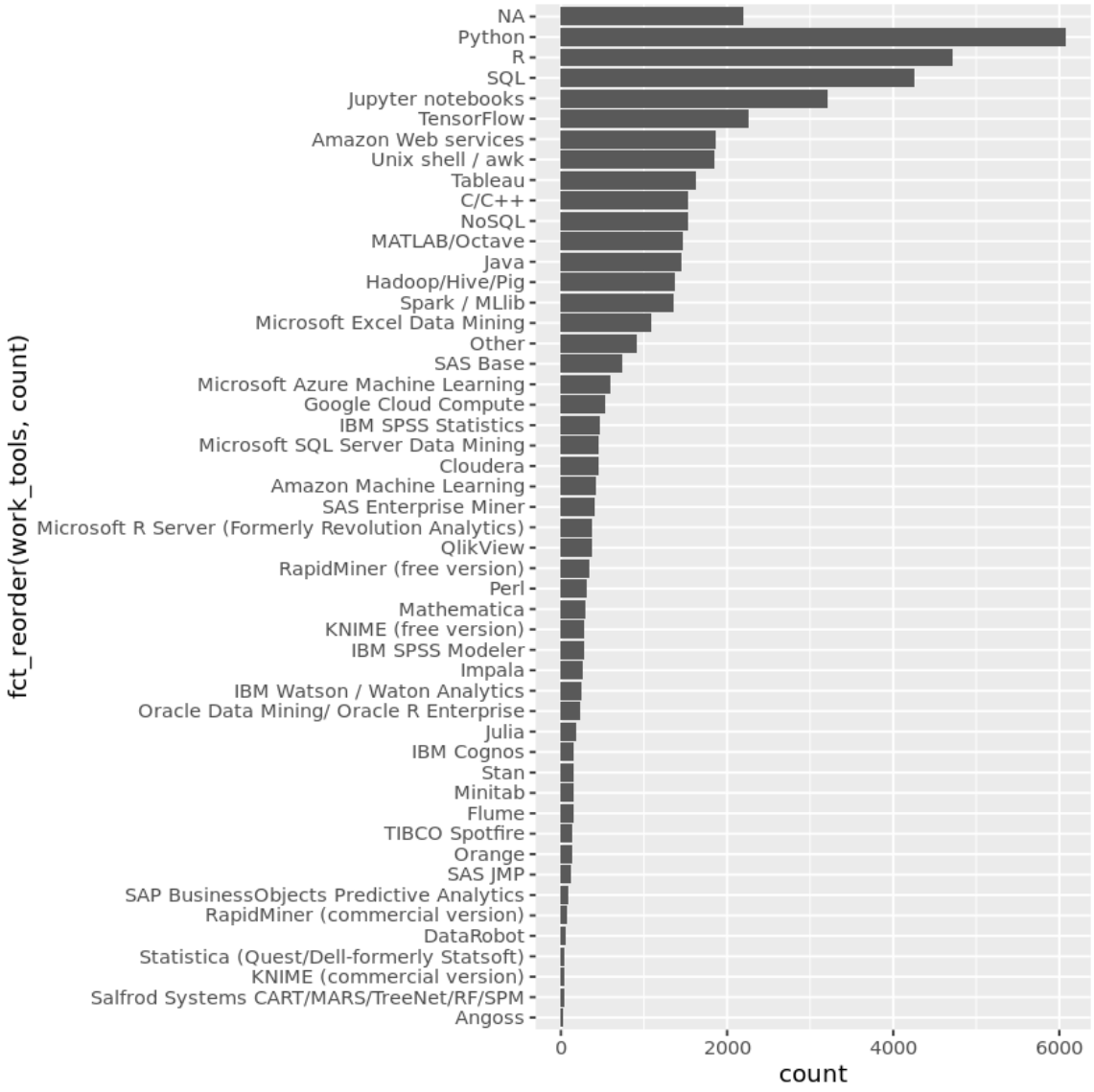
A tibble: 6 x 2

work_tools	count
<chr>	<int>
Python	6073
R	4708
SQL	4261
Jupyter notebooks	3206
TensorFlow	2256
NA	2198

4. Plotting the most popular tools

Let's see how the most popular tools stack up against the rest.

In [28]:
```
ggplot(tool_count, aes(x = fct_reorder(work_tools, count), y = count)) +
    geom_bar(stat = "identity") + coord_flip()
```



5. The R vs Python debate

Within the field of data science, there is a lot of debate among professionals about whether R or Python should reign supreme. You can see from our last figure that R and Python are the two most commonly used languages, but it's possible that many respondents use both R and Python. Let's take a look at how many people use R, Python, and both tools.

In [43]:
```
debate_tools <- responses  %>%
    mutate(language_preference = case_when(
str_detect(WorkToolsSelect, "R") & ! str_detect(WorkToolsSelect, "Python") ~
str_detect(WorkToolsSelect, "Python") & ! str_detect(WorkToolsSelect, "R") ~
str_detect(WorkToolsSelect, "R") & str_detect(WorkToolsSelect, "Python")   ~
TRUE ~ "neither"))

head(debate_tools)
```

A tibble: 6 x 6

Respondent	WorkToolsSelect	L
<dbl>	<chr>	
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	
2	Amazon Machine Learning,Amazon Web services,Cloudera,Hadoop/Hive/Pig,Impala,Java,Mathematica,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft SQL Server Data Mining,NoSQL,Python,R,SAS Base,SAS JMP,SQL,Tableau	
3	C/C++,Jupyter notebooks,MATLAB/Octave,Python,R,TensorFlow	
4	Jupyter notebooks,Python,SQL,TensorFlow	
5	C/C++,Cloudera,Hadoop/Hive/Pig,Java,NoSQL,R,Unix shell / awk	
6	SQL	

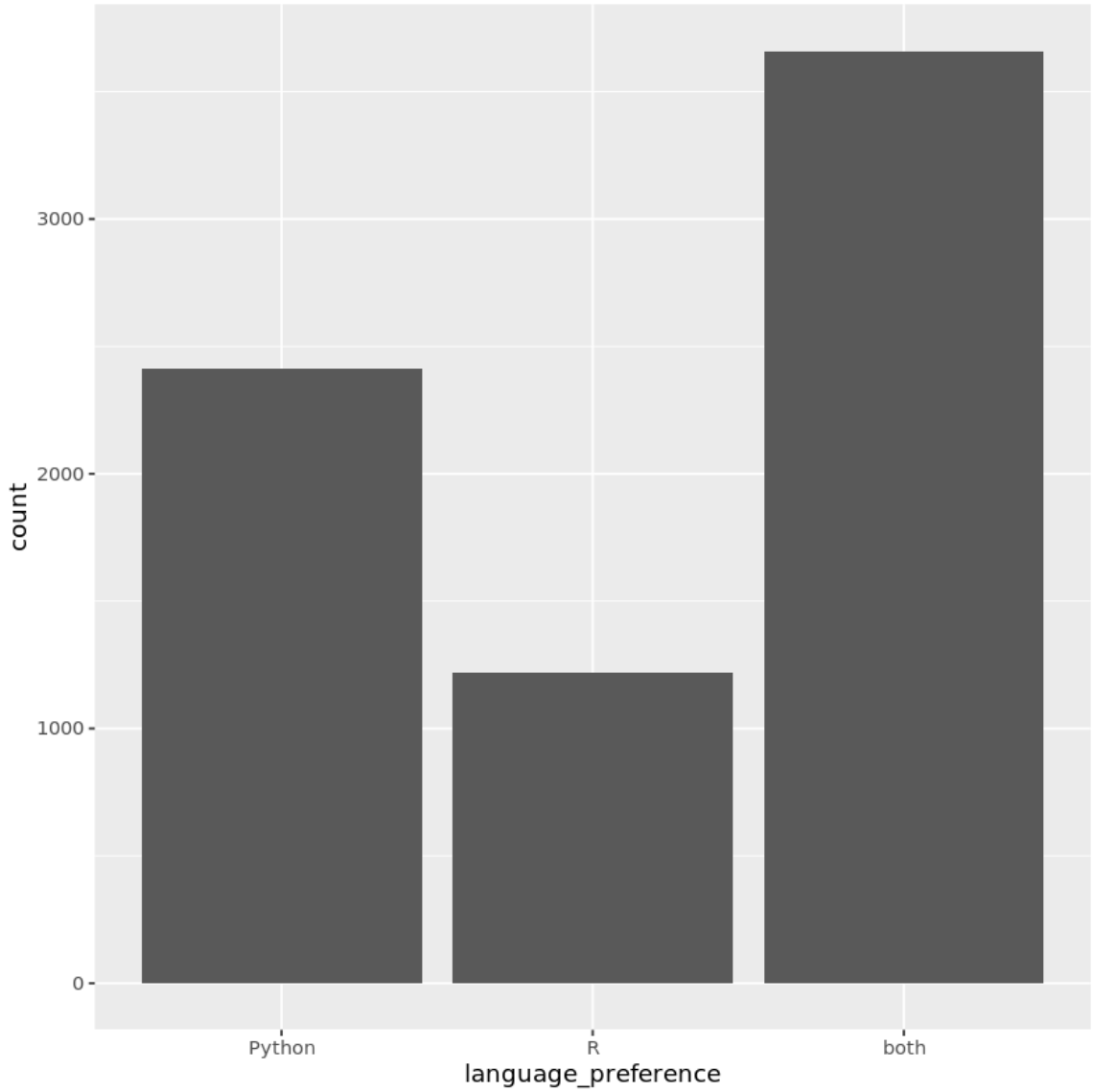
6. Plotting R vs Python users

Now we just need to take a closer look at how many respondents use R, Python, and both!

In [42]:
```
debate_plot <- debate_tools %>%
    group_by(language_preference)  %>%
    summarise(count = n())  %>%
    filter(language_preference != "neither")

ggplot(debate_plot, aes(language_preference, count)) +
        geom_bar(stat = "identity")
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```



7. Language recommendations

It looks like the largest group of professionals program in both Python and R. But what happens when they are asked which language they recommend to new learners? Do R lovers always recommend R?

In [34]:
```
recommendations <- debate_tools  %>%
    group_by(language_preference, LanguageRecommendationSelect)  %>%
    summarise(count = n())  %>%
    arrange(language_preference, desc(count))%>%
    mutate(row = row_number())%>%
    filter(row <= 4)

recommendations
```

```
`summarise()` regrouping output by 'language_preference' (override with `.
groups` argument)
```

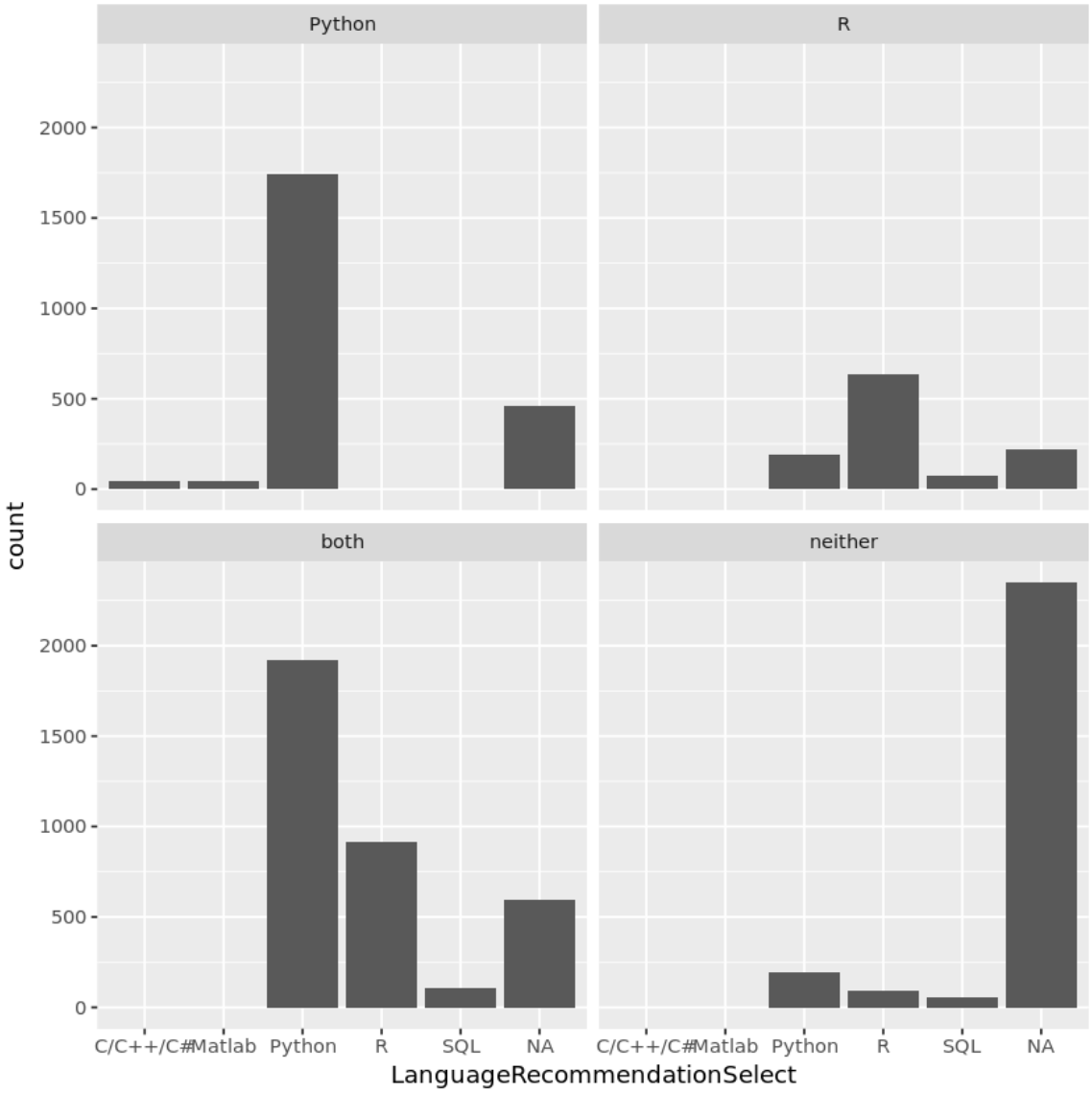
A grouped_df: 16 x 4

language_preference	LanguageRecommendationSelect	count	row
<chr>	<chr>	<int>	<int>
Python	Python	1742	1
Python	NA	459	2
Python	C/C++/C#	48	3
Python	Matlab	43	4
R	R	632	1
R	NA	221	2
R	Python	194	3
R	SQL	75	4
both	Python	1917	1
both	R	912	2
both	NA	591	3
both	SQL	108	4
neither	NA	2348	1
neither	Python	196	2
neither	R	94	3
neither	SQL	53	4

8. The most recommended language by the language used

Just one thing left. Let's graphically determine which languages are most recommended based on the language that a person uses.

In [36]:
```
ggplot(recommendations, aes(LanguageRecommendationSelect, count)) +
geom_bar(stat = "identity") + facet_wrap(~language_preference)
```



9. The moral of the story

So we've made it to the end. We've found that Python is the most popular language used among Kaggle data scientists, but R users aren't far behind. And while Python users may highly recommend that new learners learn Python, would R users find the following statement `TRUE` or `FALSE`?

In [38]:
```
# Would R users find this statement TRUE or FALSE?
R_is_number_one = TRUE
```